

Intro to Digital Design

L6: Synchronous Timing Constraints

Instructor: Naomi Alterman

Teaching Assistants:

Derek de Leuw

Isabel Froelich

Kevin Hernandez

Aarjav Jain

Packard Stephenson

Administrivia

- ❖ Lab 6 out later today - Tug of War game
 - *Bigger* step up in difficulty from Lab 5
 - Putting together complex system built of multiple FSMs – interconnections!
 - Bonus points for smaller resource usage

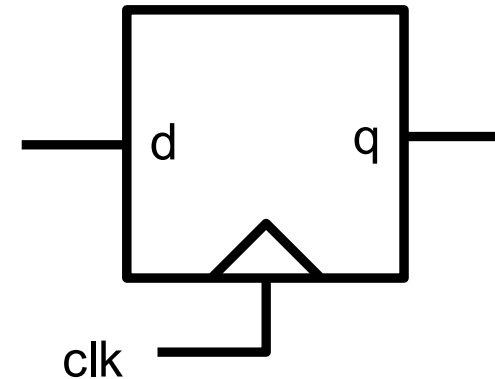
Outline

- ❖ **Synchronous Timing Constraints**

Reminder: Flip Flops

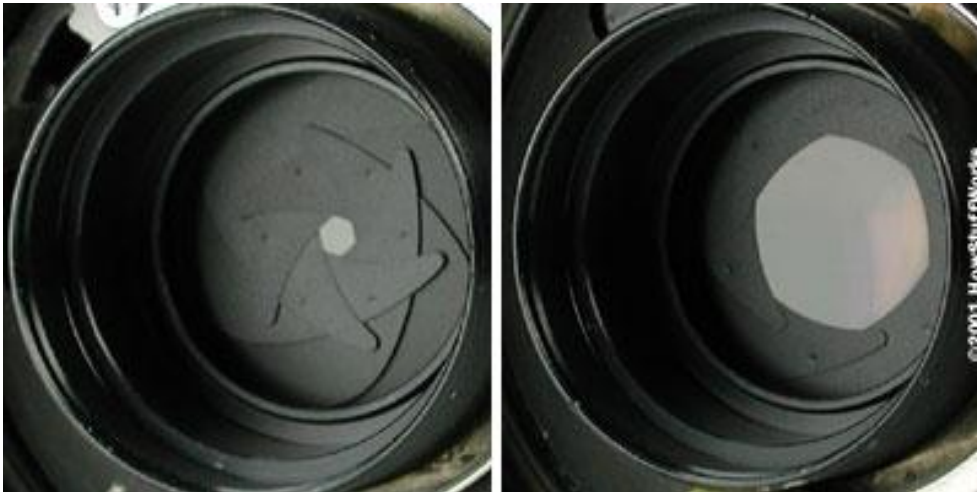
- ❖ A single bit of memory
- ❖ Copy d to q on the rising edge of the clock signal

```
module DFF (q, d, clk);  
    output logic q; // q is state-holding  
    input  logic d, reset, clk;  
  
    always_ff @(posedge clk) begin  
        q <= d;  
    end  
  
endmodule
```



Flip-Flop Timing Terminology (1/2)

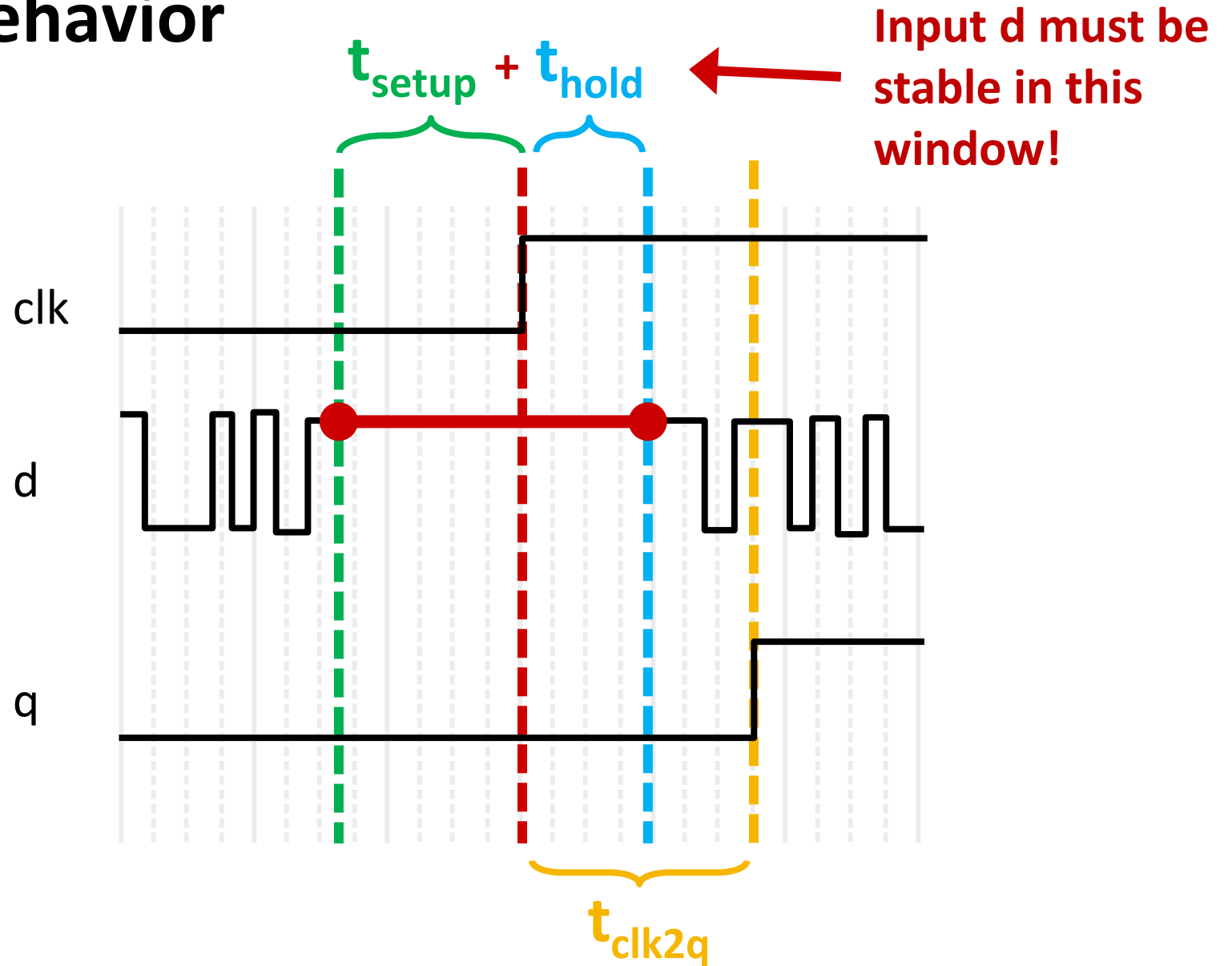
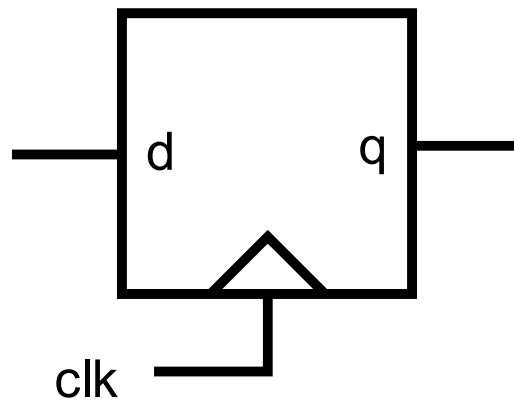
- ❖ Camera Analogy: motion blur in photos
 - *Don't move* while camera shutter is opening
 - *Don't move* while camera shutter is closing
 - *Check for blurriness* once image appears on the display



Flip-Flop Timing Terminology (2/2)

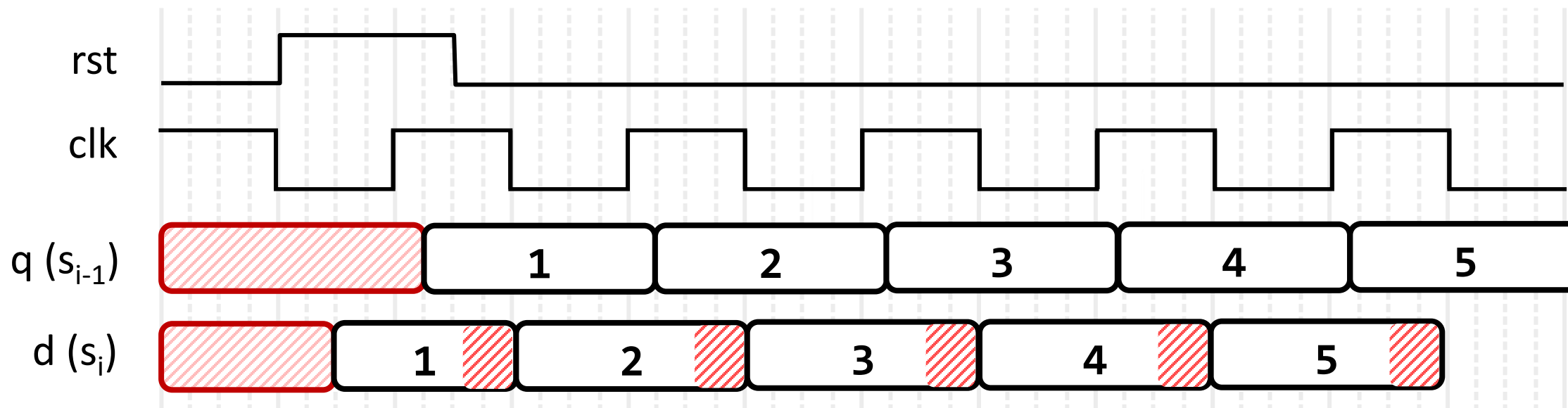
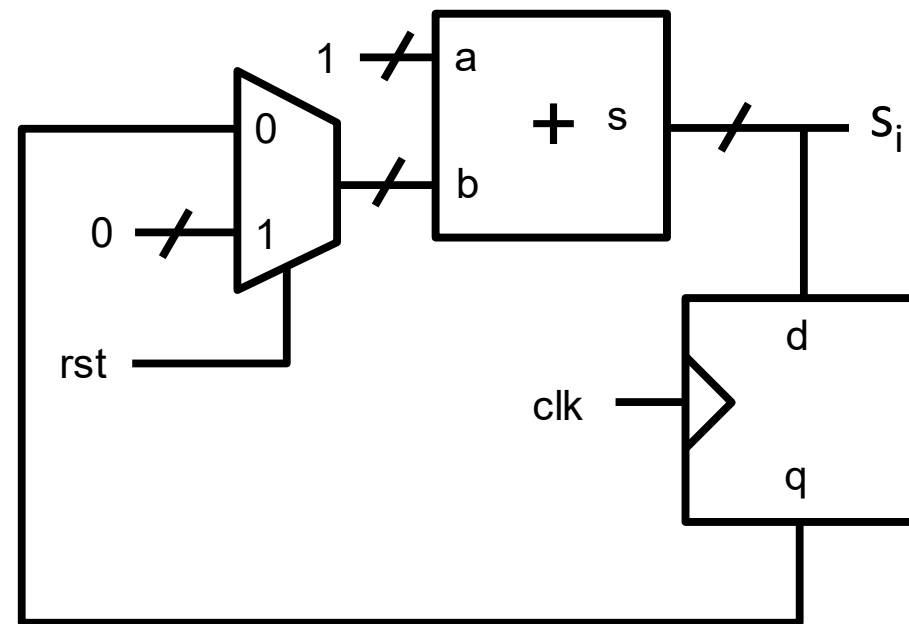
- ❖ Now applied to sequential logic elements:
 - *Setup Time*: how long the input must be stable *before* the CLK trigger for proper input read
 - *Hold Time*: how long the input must be stable *after* the CLK trigger for proper input read
 - *“CLK-to-Q” Delay*: how long it takes the output to change, measured from the CLK trigger

Flip-Flop Timing Behavior



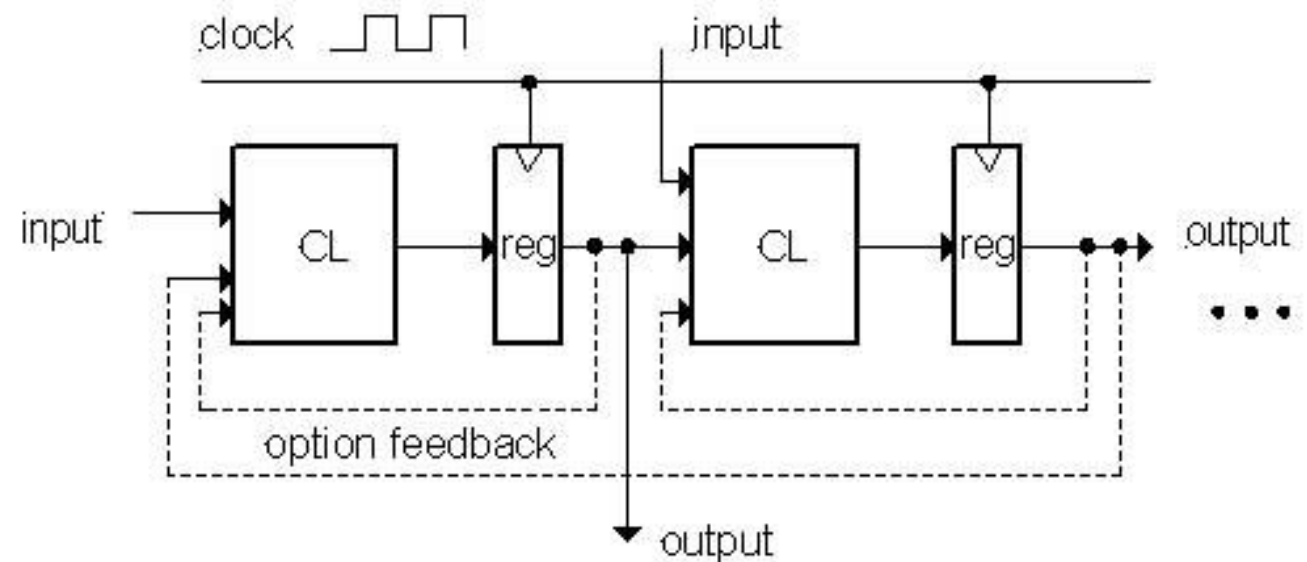
Stopwatch: timing analysis

- As bits ripple through adder, S_i is temporarily wrong!
- BUT! Register always captures correct value
- In good circuits, signal stays constant around the rising edge of CLK

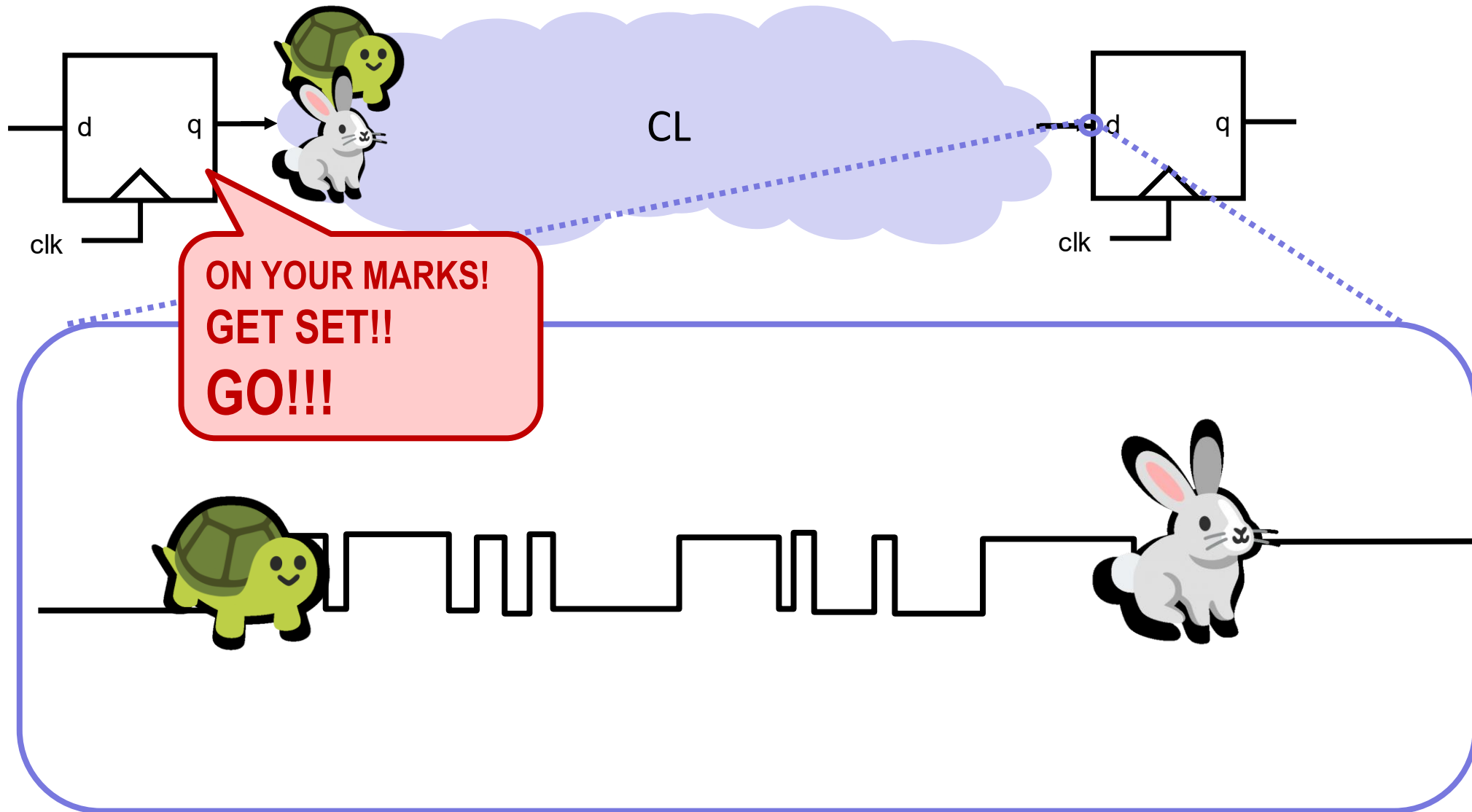


Model for Synchronous Digital Systems

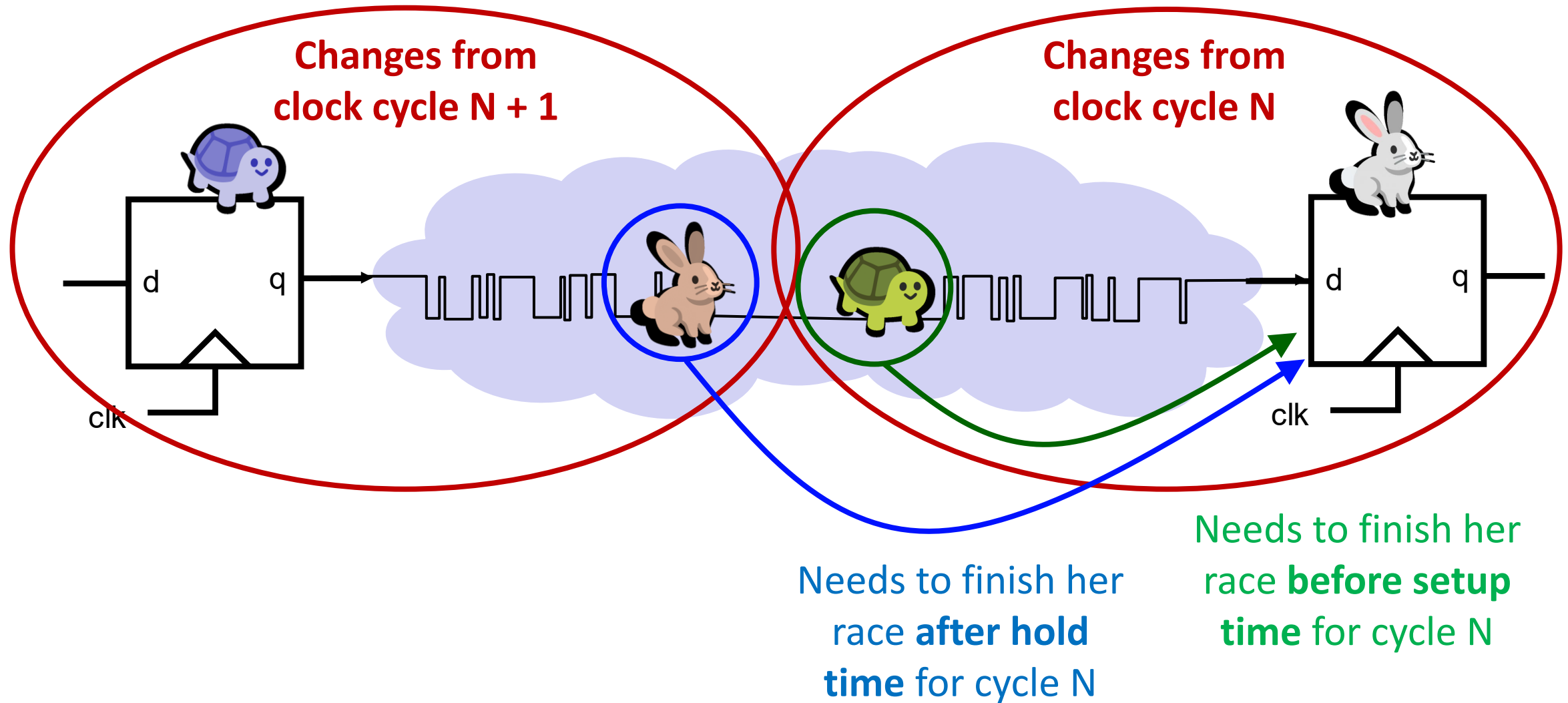
- ❖ Combinational logic blocks separated by registers
 - Clock signal connects only to sequential logic elements
 - Feedback is optional depending on application
- ❖ **How do we know if things are going to break?!**



Minimum and Maximum Delay

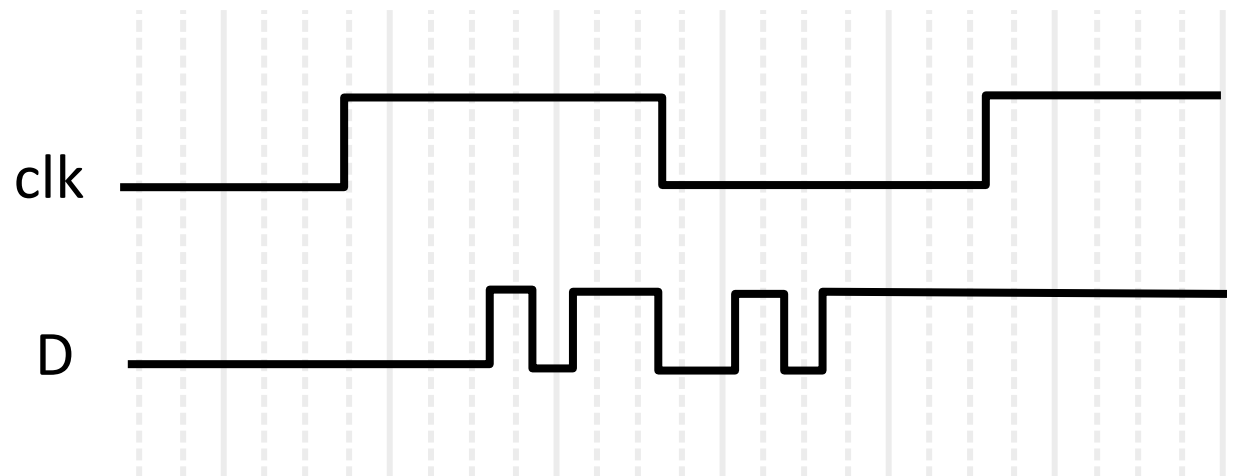


Race around the clock



Timing equations

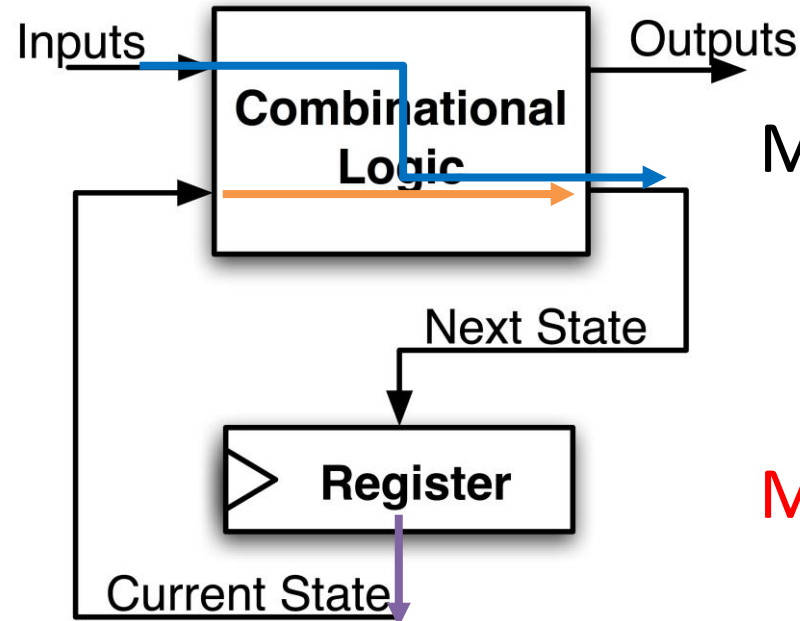
- ❖ Within a given clock period (t_{period}), the short paths 🐰 shouldn't violate hold time (t_{hold}) and the long paths 🐢 shouldn't violate setup time (t_{setup})
- ❖ Let $t_{input,i}$ be the time it takes for the input of a register to change for the i^{th} time in a single clock cycle, measured from the CLK trigger:
 - $t_{hold} \leq t_{input,i} \leq t_{period} - t_{setup}$ for all i
 - Two separate constraints!





Minimum Delay

- ❖ If shortest path to register input is too short, might violate hold time constraint
 - Input could change before state is “locked in”
 - Particularly problematic with *asynchronous* signals



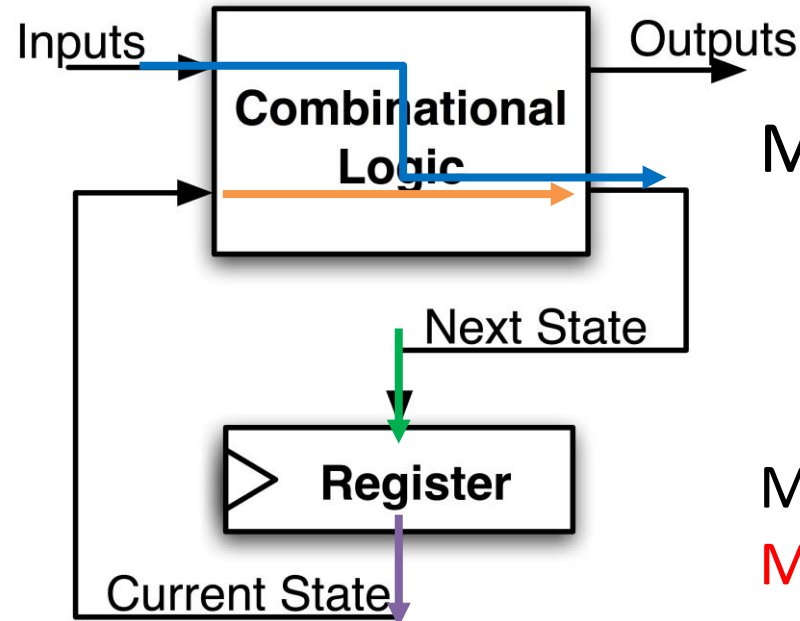
$$\text{Min Delay} = \min(\text{CLK-to-Q Delay} + \text{Min CL Delay}, \text{Min CL Delay})$$

$$\text{Min Delay} \geq \text{Hold Time}$$



Maximum Clock Frequency

- ❖ What is the max frequency of this circuit?
 - Limited by how much time needed to get correct Next State to Register (t_{setup} constraint)

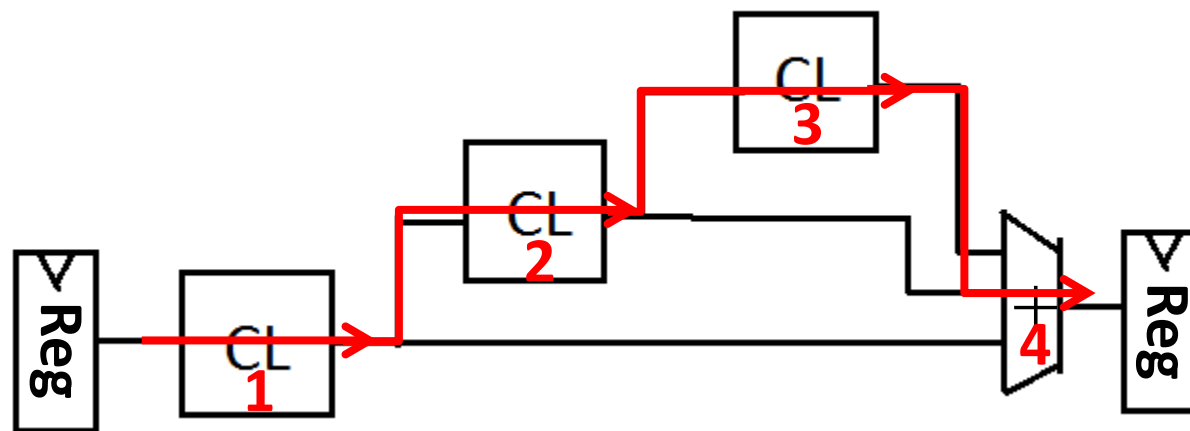


$$\text{Max Delay} = \max(\text{CLK-to-Q Delay} + \text{Max CL Delay}, \text{Max CL Delay})$$

$$\text{Min Period} = \text{Max Delay} + \text{Setup Time}$$
$$\text{Max Freq} = 1/\text{Min Period}$$

The Critical Path

- ❖ The *critical path* is the longest delay between *any* two registers in a circuit
- ❖ The clock period must be *longer* than this critical path, or the signal will not propagate properly to that next register



Critical Path =

CLK-to-Q Delay
+ CL Delay 1
+ CL Delay 2
+ CL Delay 3
+ Adder Delay
+ Setup Time

Well, what can we do then?

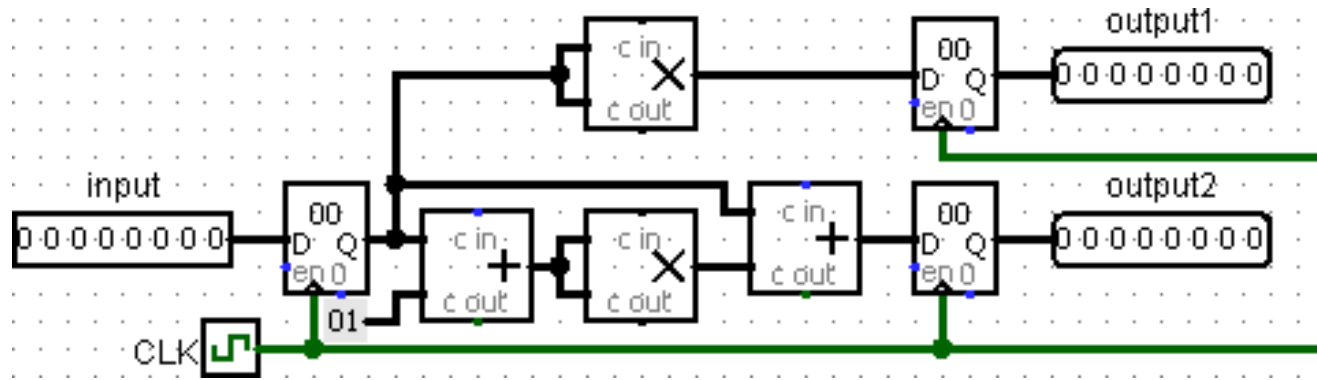
- ❖ **Choose a clock period** that's long enough to allow for the critical path
- ❖ **Design our combinational logic** so that it's neither too fast nor too slow
 - Too fast: put **delay elements** in to slow it down
 - Too slow: **split up** the logic into **two stages** with a **register in the middle**
- ❖ **Design our flip-flops** to have smaller setup, hold and clock-to-Q times

1 Practice Question

$$t_{\text{hold}} \leq t_{\text{CLmin}}$$

$$t_{\text{CLmax}} \leq t_{\text{period}} - t_{\text{setup}}$$

We want to run our CPU at 1 GHz. Assume:
 $t_{\text{add}} = 100 \text{ ps}$, $t_{\text{mult}} = 200 \text{ ps}$, $t_{\text{setup}} = t_{\text{hold}} = 50 \text{ ps}$.
 What is the maximum $t_{\text{clk-to-q}}$ we can allow?



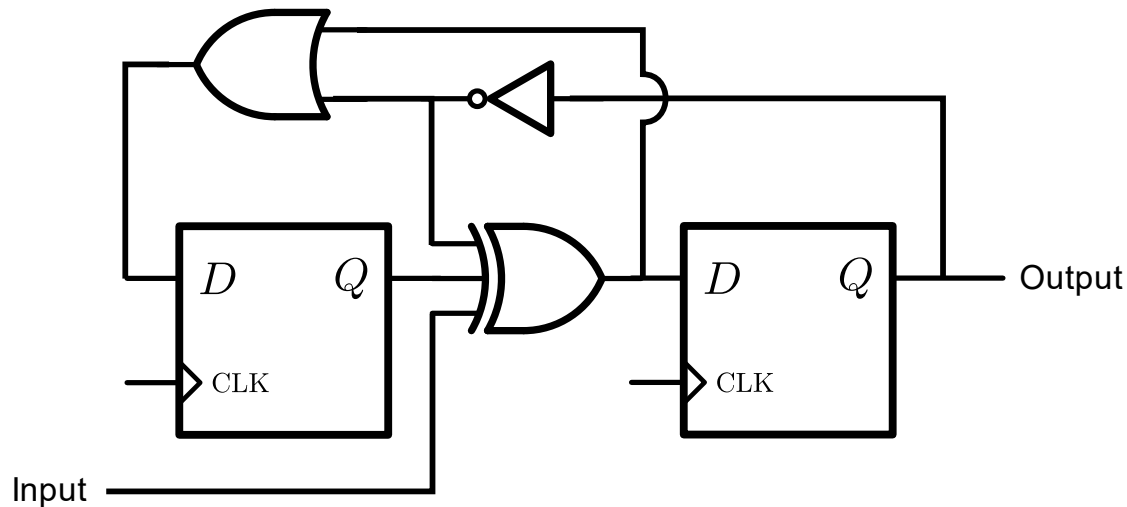
(A) 550 ps (B) 750 ps

(C) 500 ps (D) 700 ps

2 Another practice question

❖ The circuit below has the following timing parameters

- $t_{\text{period}} = 20 \text{ ns}$, $t_{\text{setup}} = 2 \text{ ns}$
- $t_{\text{XOR}} = t_{\text{OR}} = 5 \text{ ns}$, $t_{\text{NOT}} = 4 \text{ ns}$
- Input changes 1 ns after clock trigger



$$t_{\text{hold}} \leq t_{\text{CLmin}}$$

$$t_{\text{CLmax}} \leq t_{\text{period}} - t_{\text{setup}}$$

1. What is the max allowable t_{C2Q} ?

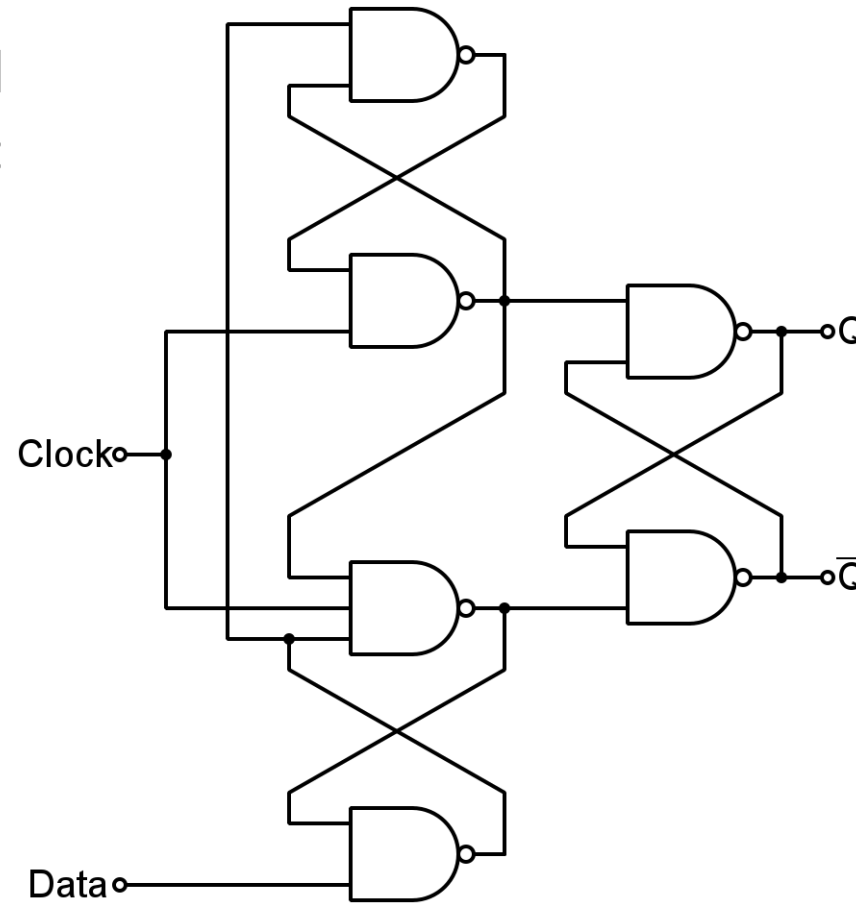
2. If $t_{\text{C2Q}} = 3 \text{ ns}$, what is the max allowable t_{hold} ?

Miso Moment



Where Do Timing Constraints Come From?

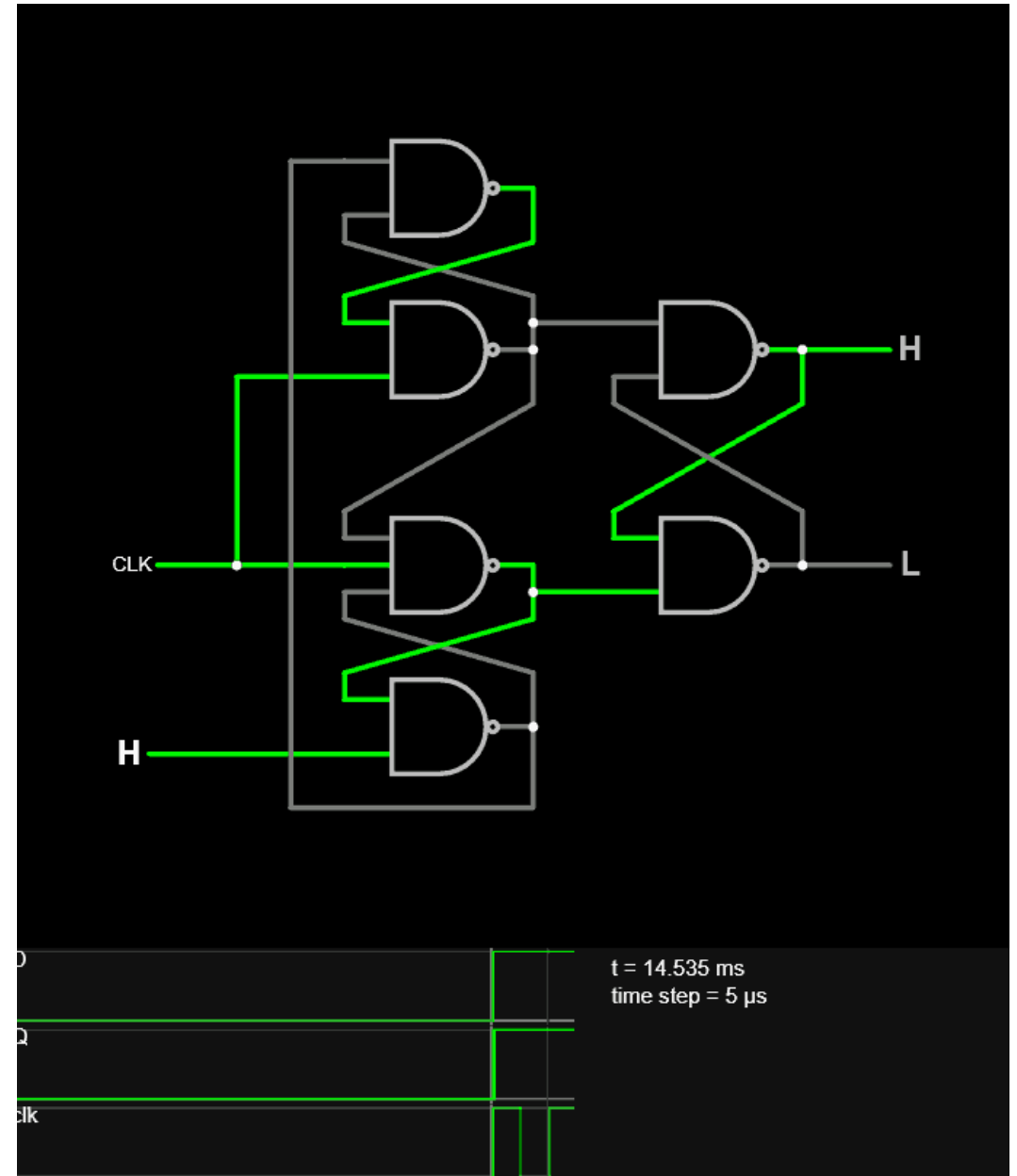
Edge-triggered
D flip-flop:



By Nolanjshettle at English Wikipedia, CC BY-SA 3.0,
<https://commons.wikimedia.org/w/index.php?curid=40852354>

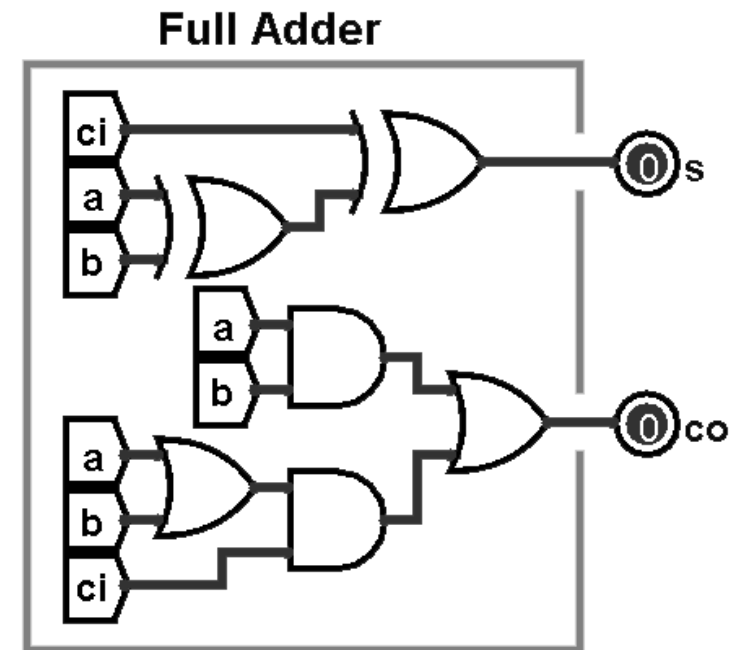
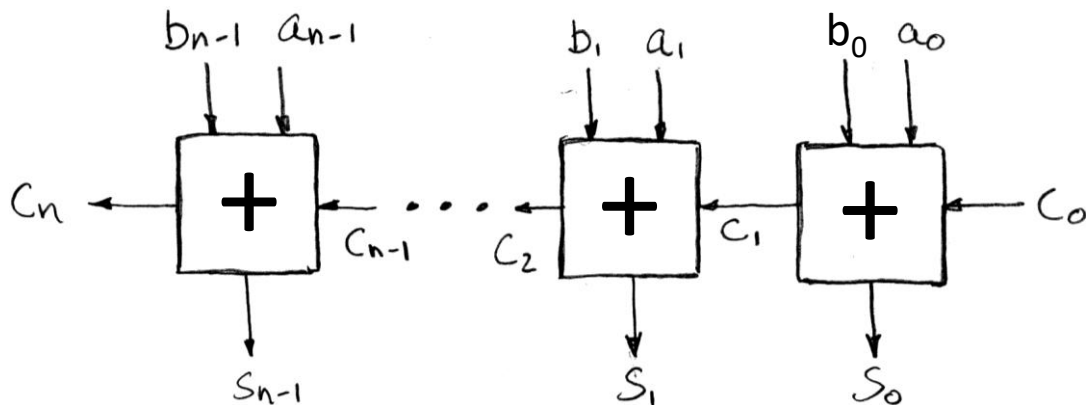
DFF feedback loop simulator

- ❖ <https://www.falstad.com/circuit/e-edgedff.html>
- ❖ This is **way** outside the scope of this class. Only worry about this if you're curious



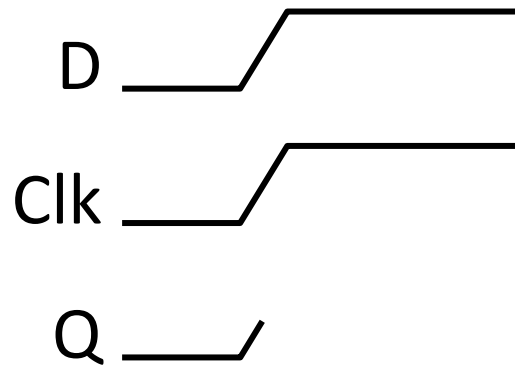
3 More timing Practice

- ❖ For an n -bit ripple-carry adder, what is the **earliest** and **latest** time that the output will change after clock edge?
 - Assume gate delay of 1 ns
 - Assume A, B, c_0 straight from register output (eg, change at t_{C2Q})



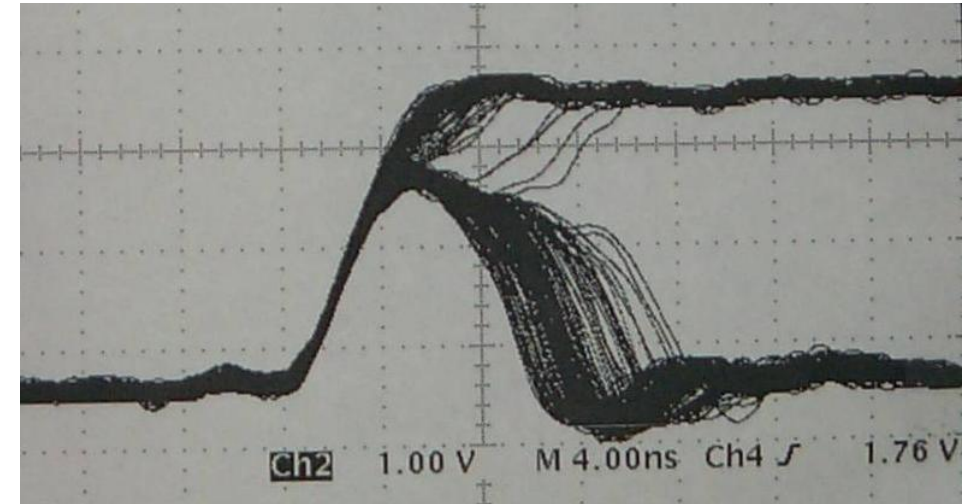
Flip-Flop Realities: External Inputs

- ❖ External inputs aren't synchronized to the clock
 - If not careful, can violate timing constraints
- ❖ What happens if input changes around clock trigger?

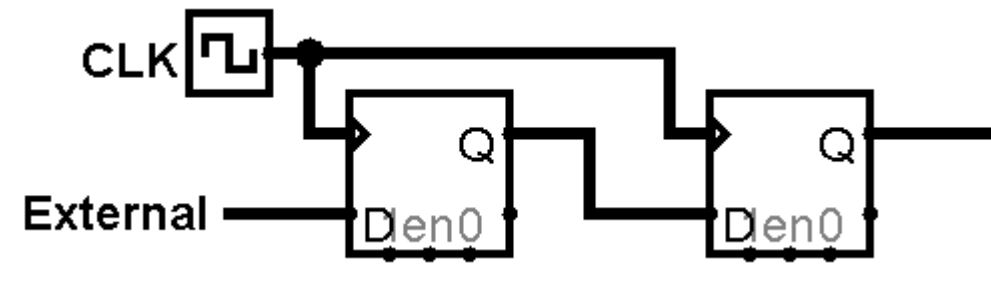


Flip-Flop Realities: Metastability

- ❖ **Metastability** occurs when a digital feedback loop settles into an **unstable equilibrium** storing a **non-binary** voltage
 - Can last for a potentially unbounded amount of time
 - Will randomly decay to a '0' or a '1'probably
- ❖ Flip flops help reject transients
 - Longer chains = higher chance of filtering out all metastability, but longer signal delay

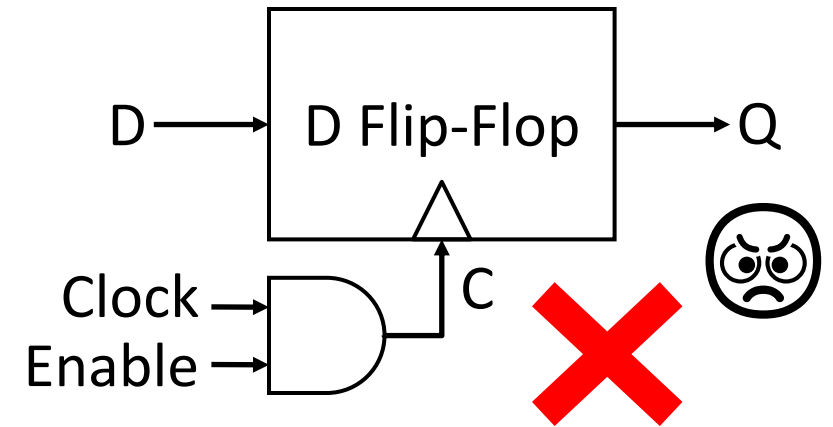


<https://www.cl.cam.ac.uk/teaching/1011/SysOnChip/slides/sp3soc/parts/zhp6e41885b8.html>



Flip-Flop Realities: **NEVER GATE THE CLOCK!!!**

- ❖ Don't do it
- ❖ Introduces unnecessary **clock skew** into design
- ❖ Makes timing equations ***exceptionally*** hard to model



Summary of Timing Terms

- ❖ **Clock:** steady square wave that synchronizes system
- ❖ **Flip-flop:** one bit of state that samples every rising edge of CLK (positive edge-triggered)
- ❖ **Register:** several bits of state that samples on rising edge of CLK (positive edge-triggered); often has a RESET
- ❖ **Setup Time:** when input must be stable *before* CLK trigger
- ❖ **Hold Time:** when input must be stable *after* CLK trigger
- ❖ **CLK-to-Q Delay:** how long it takes output to change from CLK trigger

Extra: Unrelated to SDS, just more FSM practice

❖ Recognize the string 101 with the following behavior

■ Input: 1 0 0 1 0 1 0 1 1 0 0 1 0

■ Output: 0 0 0 0 0 1 0 1 0 0 0 0 0

❖ State diagram to implementation:

